AF IZW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:

Allavarpu, et al.

Serial No. 09/557,068

Filed: April, 21, 2000

For: IDL Event and Request Formatting for CORBA Gateway

§ Group Art Unit: 2154
§
§ Examiner: Hu, Jinsong
§
§ Atty. Dkt. No.: 5181-61100
§ P5026

CERTIFICATE OF MAILING
37 C.F.R. § 1.8

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below:

Robert C. Kowert
Name of Registered Representative

February 18, 2005
Date

Signature

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal filed December 15, 2004, Appellants present this Appeal Brief. Appellants respectfully request that the Board of Patent Appeals and Interferences consider this appeal.

I. REAL PARTY IN INTEREST

As evidenced by the assignment recorded at Reel/Frame 010993/0702, the subject application is owned by Sun Microsystems, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and now having its principal place of business at 4150 Network Circle, Santa Clara, CA 95054.

II. RELATED APPEALS AND INTERFERENCES

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-45 are pending and rejected. The rejection of claims 1-45 is being appealed. A copy of claims 1-45 is included in the Claims Appendix hereto.

IV. STATUS OF AMENDMENTS

No amendments to the claims have been submitted subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Many types of devices may be managed over a network, such as printers, scanners, phone systems, copiers, and many other devices and appliances configured for network operation. Typically, such devices are managed via requests and events. A request is sent in a message to a managed object. For example, a request may be sent by a manager application to a managed object to query the object about a particular parameter associated with the object. A request may also be sent to a managed object to modify a parameter of the object. Alternately, an event is a message originating with a

managed object. Events may be sent by managed objects to signal some change of state of the managed object, or to communicate information about the managed object. Typically, network management software manages a given device by storing and manipulating a representation of its pertinent data as a software object, which may be referred to as a "managed object." This object may be a virtual representation of the device on the network.

Claim 1 is directed to a network management system in which a gateway delivers messages between managed objects and managers. For instance, such a network management system may include a CORBA gateway delivering messages between CORBA-based appellants and an enterprise manager, as described in the Specification on page 11, lines 4-25. A CORBA gateway may translate manager requests, such as from IDL (Interface Definition Language) requests to PMI (Portable Management Interface) requests. A gateway may include other gateway components, such as a request gateway or an event gateway. An event gateway may deliver events, such as notifications, warnings or alarms, from managed objects to manager objects or clients. For instance, an event gateway may collect and filter events from managed objects, covert and deliver them to clients. *See, e.g.*, FIG. 3, and 4; page 11, line 27 – page 12, line 12; and page 20, line 27 – page 24, line 11; page 25, line 12 – 26, line 11.

Similarly, requests from clients or managers may be delivered to managed object via a request gateway. A request gateway may translate requests, such as from IDL to PMI, and deliver the requests to managed objects. A request gateway may also deliver responses from managed objects back to the requesting client or manager object. *See, e.g.*, FIGs. 7, 8, 9, page 13, lines 1-16; page 22, lines 16 – 27; page 24, lines 20-30; page 28, line 26 – page 30, line 18.

Additionally, the gateway in the network management system may deliver messages between managers and managed objects using a platform-independent interface and in a format selected by the manager. For example, IDL APIs may be provided to allow a client or manager object to choose the format, such as text or ASN.1, in which

events are delivered to the client. *See, e.g.*, FIGs. 1a, 2, 3, 7, 8, 10, 11,13 and 15; page 11, line 27 – page 12, line 12; page 22, lines 7-13; page 36, line18-page 37, line23.

Claims 16 and 31 are directed, respectively, to a method and a medium comprising program instructions executable to implement or perform the functions of a network management system as described above regarding independent claim 1.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-3, 5, 6, 16-18, 20, 21, 31-33, 35 and 36 stand finally rejected under 35 U.S.C. § 102(e) as being anticipated by Carre (U.S. Patent 6,282,579).

2. Claims 1, 2, 4-11, 13-17, 19-26, 28-32, 34-41 and 43-45 stand finally rejected under 35 U.S.C. § 102(e) as being anticipated by Shank et al. (U.S. Patent 6,445,776) (hereinafter "Shank").

3. Claims 3, 12, 18, 27, 33 and 42 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Shank.

VII. ARGUMENT

First Ground of Rejection:

Claims 1-3, 5, 6, 16-18, 20, 21, 31-33, 35 and 36 are rejected under 35 U.S.C. § 102(e) as being anticipated by Carre (U.S. Patent 6,282,579). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 1, 3, 5 and 6:

Carre does not anticipate a gateway configured to deliver messages between managed objects and one or more managers through a platform-independent interface,

wherein the gateway is configurable to deliver the messages for each manager in a format selected by that manager. Instead, Carre pertains to address conversion between CORBA objects and OSI objects (Carre - col. 1, lines 9-19; col. 1, line 59 - col. 2, line 46) and to the transforming of object interfaces (column 5, lines 49-59). Thus, Carre is concerned with converting *address types and object interfaces*, but fails to disclose anything regarding message formats and a gateway that is configurable to deliver the messages for each manager in a format selected by that manager.

Specifically, Carre teaches that address conversion is performed *according to the types* of objects that are communicating. Carre's managers cannot select a desired message format. There is clearly no such functionality described in Carre. The sections cited by the Examiner (col. 5, lines 49-59 and col. 6, lines 30-35) refer to address-type conversion between CORBA objects and OSI objects. There is absolutely no mention in Carre of managers being able to select the format for messages delivered by the gateway. Nor does Carre does not describe any mechanism by which a manager can select a format for messages. Carre fails to mention anything about different message formats. The gateway in Carre is clearly not described as being capable of allowing the managers to select a format.

In response to above arguments, the Examiner refers to Carre's teachings regarding the delivery of messages through different interfaces (CDMO and CMISE) by gateways and cites Figures 3a and 3b. The Examiner contends that since Carre teaches more than one gateway and since they each communicate via different interfaces, they perform the same function as a gateway configurable to deliver messages for each manager in a format selected by that manager. However, the Examiner's interpretation of Carre's interfaces is incorrect. Specifically, Carre states that his interface units translate an interface to the underlying object so that the underlying object "can be accessed by classic CORBA messages" (Carre, column 5, lines 50-52). Carre also states that his CMISE/IDL interface appears to the outside like a CORBA object (Carre, column 5, lines 26-31).

One portion of Carre cited by the Examiner (column 5, lines 49-59) describes how OSI objects OM and OA can be *transformed into pure CORBA objects* to allow them to be accessed using classic CORBA messages. Thus, Carre is clearly teaching the transformation of object interfaces so that a single message format (i.e. classic CORBA) may be used with either OSI objects or CORBA objects. Furthermore, Carre's manager objects cannot select which interface to communicate through. On the contrary, Carre teaches that his interfaces are present to allow interaction between CORBA and OSI objects "via a CORBA infrastructure" (Carre, column 4, line 63 – column 5, line 3). Carre teaches the use of additional communication layers (GDMO/C++, GDMO/IDL and CMISE/IDL), or components, between OSI objects and CORBA objects that translate the interfaces to the objects such that they appear as, and are accessible as, CORBA objects using CORBA messages (Carre, Figures 2a and 2b, column 5, lines 49-59). In other words, Carre's interfaces are present specifically to provide communication between otherwise incompatible objects.

Furthermore, Carre's interfaces are not message formats. Even if one of Carre's managers could select a different interface, which Appellants maintain they cannot, such a selection would still not be selecting a format for message delivery as the Examiner contends. Additionally, if one of Carre's managers could select a different message format, that manager would not be able to interact with a target object, which because of Carre's interface transformations is accessible via classic CORBA messages. Object interfaces and message formats are different things. In Carre's invention, different interfaces are provided specifically to allow different object types (specifically OSI or non-CORBA object) to access and send message through a single CORBA infrastructure. Carre's system is quite different from a gateway configured to deliver messages for managers in *formats selected by the managers*.

Carre specifically teaches the use of object interfaces and additional communication layers to allow heterogeneous (CORBA and OSI) objects to communicate via a single, homogeneous infrastructure (CORBA), rather than having a gateway configurable to deliver messages in formats selected by managers. Thus, Carre

is teaching away from a gateway configured to deliver messages for managers in formats selected by those managers.

For a proper rejection under § 102, the identical invention must be shown in as complete detail as is contained in the claims. M.P.E.P. § 2131; *see also, Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). Carre does not disclose managers able to select message formats, nor does Carre teach a gateway (or any other mechanism) through which a manager could make such a selection. Thus, Carre clearly does not anticipate a gateway configurable to deliver messages for each manager in a format selected by that manager.

Claim 2:

In regard to claim 2, Carre does not anticipate that the selected format comprises text. The Examiner cites column 6, lines 30-35 of Carre. However, this passage of Carre merely mentions the mapping of ASN.1 onto IDL data types to allow translation of OSI *address types* to CORBA address types. Additionally, Carre is not discussing the mapping of ASN.1 to IDL data types or the translation of address types from OSI to CORBA in the context of a format selected by a manager for message delivery. Instead, Carre is talking about the translations necessary to allow OSI objects to communicate via classic CORBA. Further, Carre does not teach that such address translations involve a format that comprises text.

In response to Appellants' arguments, the Examiner cites the teaching of Carre for sending the outcome message to the client based on information required by the client in a request message and further argues that "[a]ll of these messages include context and [are] related to different target object[s]." However, including a request context in a request message does not disclose (or teach or suggest) a client selecting a message

format comprising text. In fact, Carre fails to mention anything about message formats comprising text. Carre simply teaches that request messages can include: an operation, a target object, one or more parameters, and, optionally a request context. Clients including a request context in a request message, as taught in Carre, has nothing to do with a manager selecting a delivery format comprising text.

Claims 16, 18, 20 and 21:

Carre does not anticipate one or more managers each selecting a format for messages deliverable by a gateway between one or more managed objects and each of the one or more managers, as the examiner contends. As noted above regarding claim 1, Carre pertains to address conversion between CORBA objects and OSI objects (*see,* Carre - col. 1, lines 9-19; col. 1, line 59 - col. 2, line 46) and to the transformation of object interfaces column 5, lines 49-59). Thus, Carre is concerned with converting *address types and object interfaces*, but fails to disclose anything regarding message formats.

Carre's address conversion is performed according to the types of objects that are communicating. There is no ability in Carre for the managers to select a desired message format. The sections cited by the Examiner (col. 5, lines 49-59 and col. 6, lines 30-35) refer to address-type conversion between CORBA objects and OSI objects. There is absolutely no mention in Carre of managers being able to select the format for messages delivered by the gateway. Nor does Carre does not describe any mechanism by which a manager can select a format for messages. Carre fails to mention anything about different message formats. The gateway in Carre is clearly not capable of allowing the managers to select a format.

Please see the arguments above regarding claim 1 for a more detailed discussion of Carre's failure to disclose the ability for managers to select a format for the delivery of messages between managers and managed objects. In summary, Carre teaches only transforming interfaces, which are clearly not messages or message formats. Carre

specifically teaches the use of object interfaces and additional communication layers to allow heterogeneous (CORBA and OSI) objects to communicate via a single, homogeneous infrastructure (CORBA), rather than having a gateway configurable to deliver messages in formats selected by managers. Thus, Carre is teaching away from a gateway configured to deliver messages for managers in formats selected by those managers.

Claim 17:

In regard to claim 17, Carre does not teach that the selected format comprises text. The Examiner cites column 6, lines 30-35 of Carre. However, this passage of Carre merely mentions the mapping of ASN.1 onto IDL data types to allow translation of OSI address types to CORBA address types. Additionally, Carre is not discussing the mapping of ASN.1 to IDL data types or the translation of address types from OSI to CORBA in the context of a format selected by a manager for message delivery. Instead, Carre is talking about the translations necessary to allow OSI objects to communicate via CORBA. Further, Carre does not teach that such address translations involve a format that comprises text.

Additionally the arguments presented above regarding claim 2 apply to claim 17 with equal force. Please see the above arguments regarding claim 2 for a more detailed discussion regarding Carre's failure to disclose a selected format comprising text.

Claim 31, 33, 35 and 36:

Regarding claim 31, Carre does not disclose one or more managers each selecting a format for messages deliverable by a gateway between one or more managed objects and each of the one or more managers. As noted above regarding claims 1 and 16, Carre pertains to address conversion between CORBA objects and OSI objects (*see,* Carre - col. 1, lines 9-19; col. 1, line 59 - col. 2, line 46) and to the transformation of object interfaces column 5, lines 49-59). Thus, Carre is concerned with converting *address types and object interfaces*, but fails to disclose anything regarding message formats.

Carre address conversion according to the types of objects that are communicating. There is no ability in Carre for the managers to select a desired message format. The sections cited by the Examiner (col. 5, lines 49-59 and col. 6, lines 30-35) refer to address-type conversion between CORBA objects and OSI objects. There is absolutely no mention in Carre of managers being able to select the format for messages delivered by the gateway. Nor does Carre does not describe any mechanism by which a manager can select a format for messages. Carre fails to mention anything about different message formats. The gateway in Carre is clearly not described as allowing the managers to select a format.

Please see the arguments above regarding claims 1 and 16 for a more detailed discussion of Carre's failure to disclose the ability for managers to select a format for the delivery of messages between managers and managed objects. In summary, Carre teaches only transforming interfaces, which are clearly not messages or message formats. Carre specifically teaches the use of object interfaces and additional communication layers to allow heterogeneous (CORBA and OSI) objects to communicate via a single, homogeneous infrastructure (CORBA), rather than having a gateway configurable to deliver messages in formats selected by managers. Thus, Carre is teaching away from a gateway configured to deliver messages for managers in formats selected by those managers.

Claim 32:

In regard to claim 32, Carre does not teach that the selected format comprises text. The Examiner cites column 6, lines 30-35 of Carre. However, this passage of Carre merely mentions the mapping of ASN.1 onto IDL data types to allow translation of OSI address types to CORBA address types. Additionally, Carre is not discussing the mapping of ASN.1 to IDL data types or the translation of address types from OSI to CORBA in the context of a format selected by a manager for message delivery. Instead, Carre is talking about the translations necessary to allow OSI objects to communicate via

CORBA. Further, Carre does not teach that such address translations involve a format that comprises text.

Additionally the arguments presented above regarding claims 2 and 17 apply to claim 17 with equal force. Please see the above arguments regarding claims 2 and 17 for a more detailed discussion regarding Carre's failure to disclose a selected format comprising text.

Second Ground of Rejection:

Claims 1, 2, 4-11, 13-17, 19-26, 28-32, 34-41 and 43-45 are rejected under 35 U.S.C. § 102(e) as being anticipated by Shank et al. (U.S. Patent 6,445,776) (hereinafter "Shank"). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 1, 4, 5, 6, 7, 8, 10, 14 and 15:

Shank does not anticipate a network management system comprising a gateway configured to deliver messages between managed objects and one or more managers through a platform-independent interface, wherein the gateway is configurable to deliver the messages for each manager in a format selected by that manager. Instead, Shank pertains to providing telephony and media services from a server 110 to an application 140 (Shank, Figure 1, column 1, lines 13-18). Shank is not concerned with, nor does Shank pertain to, managers and managed objects. According to Shank, a server may include various service interfaces, such as telephony services 210, media services 220, and basic services 230 that a client may use. Shank's system provides a CORBA ORB 260 for communicating with these interfaces (col. 3, line 31 - col. 4, line 13). As described in Shank, the service interfaces (such as telephony services 210 and media services 220) allow client application 140 to interact with services such as telephone services provided on telephone network 105 and media services provided by various hardware components (col. 7, lines 15-28).

Contrary to the Examiner's assertions, the service interfaces 210, 220 and 230 of Shank's server 110, do not provide a gateway configurable to deliver the messages for each manager in a format selected by that manager. Shank does not pertain to interactions between managers and managed objects as these entities are understood in the art. Instead, Shank only discusses the client-server interactions between application 140 and server 110. In other words, Shank only discusses providing telephony and media services through a server to a client application. As discussed above, Shank's interfaces 210, 220, 230 provide service interfaces for an application 140. They do not deliver messages between managed objects and one or more managers. Telephony service interface 210 is not a manager for managed objects. The concept of managers and managed objects, and the relationship between managers and managed objects, is well understood in the art of managed networks. In contrast, telephony service interface 210 (including 212-216) is clearly described in Shank as providing an interface for application 140 to access services on telephony network 105. Thus, interfaces 210-216 in Shank are not described as having anything to do with managing managed objects on a managed network.

Furthermore, Shank clearly does not teach a gateway that is configurable to deliver the messages for each manager in a format selected by that manager. The Examiner refers to col. 5, lines 39-50 and col. 17, lines 26-37. However, these portions of Shank merely give examples of media and telephony services accessible through Shank's interfaces 220 and 210. This portion of Shank has nothing to do with message formats, let alone delivering a message in a format selected by a manager. The Examiner's cited references have no relevance to managers selecting message formats. The Player, Recognizer, etc. discussed in Shank are media services, not managers for managed objects in a managed network. Moreover, there is clearly no teaching in Shank of a manager using any these services to select a format for message delivery.

Additionally, Shank teaches that his service interfaces are defined according to the target object or target hardware, such as text-to-speech services 222, or facsimile services 228, and fails to teach that the formats of messages are selected by a manager

managing such a target object. In fact, Shank teaches the use of a custom format "based on similar methods specified in the ECTF S 1.00API," but defined using IDL (Shank, column 17, lines 31-34). Data used by these interfaces is "in the form of a key value set (KVS) which contains a sequence of keys associated with values " and "[s]tructurally, a KVS is a sequence of key value pairs (KVPairs)" (Shank, column 9, lines 1-7). According to Shank, application 140, which the Examiner has erroneously characterized as a manager, communicates with various services using whatever interface the service has registered with resource administrator 236 (Shank, column 5, lines 16-22). Shank does not teach that application 140 selects a message format when communicating with services. Thus, Shank clearly teaches the use of predefined message formats and not the use of formats selectable by a manager.

In response to the above arguments, the Examiner refers to Shank's teaching regarding "providing services through media, telephony and basic services interfaces" and further argues that Shank's interfaces perform a message delivery function as a gateway. However, the Examiner's interpretation of Shank is incorrect. As described above, Shank's interfaces are *defined according to the target object or target hardware*, such as text-to-speech services 222, or facsimile services 228, and the *formats of messages are not selected by a manager managing* such a target object. Furthermore, as with the rejection of claim 1 over Carre, the Examiner seems to be confusing interfaces with message formats. Different interfaces do not imply selectable delivery formats. Appellants further point out that the Examiner's cited portions of Shank (column 5, lines 39-50 and column 17, lines 26-37) teach only that different interfaces may include different *method* definitions, but fail to teach anything regarding the format of messages and further fail to teach *message formats selectable by a manager.*

Claim 2:

In regard to claim 2, Shank does not disclose wherein the selected format comprises text. The Examiner cites item 228 of Figure 2. However, item 228 refers to a FAX service, and it is well known that a facsimile interface does not include text, but

rather involves a graphic interface. Appellants fail to see the relevance of item 228 of Figure 2 to a selected format that comprises *text*.

In response to the above arguments, the Examiner cites Shank's teachings regarding providing text-to-speech services. The text-to-speech services referred to by the Examiner are services accessed by client application 140. For example, the text-to-speech service converts text data supplied by application 140 into an audio file. When discussing text-to-speech, Shank is referring to a high level function performed by the service, not an inter-object message format used for communicating with the service. Even if Shank's application 140 were to represent a manager object, which Appellants assert it does not, Shank still does not teach that application 140 may choose text-to-speech as a format for message delivery when communicating with a service object. In fact, it is clear that when communicating with a text-to-speech service, one must provide the text to be converted into audio. Shank teaches that application 140 must use an interface specified by the text-to-speech service's ORB vendor (Shank, column 3, line 65-column 4, line 6). Thus, Shank's text-to-speech service is not a selectable message format for communicating between a manager and a managed object. Shank clearly does not describe a manager being able to select a format for message delivery that comprises text.

Claim 10:

In regard to claim 10, Shank does not anticipate a gateway comprising a request gateway which is configured to deliver messages generated by the one or more managers to the one or more managed objects, wherein the messages comprises requests and wherein the requests comprise a query for information concerning one of the managed objects. The portions of Shank cited by the Examiner (column 2, lines 64-67; column 7, lines 43-46; and column 7, line 66 through column 8, line 6) refer to application 140 invoking functions of the telephony and media services. These teachings have nothing to do with a query for information concerning a managed object. The concepts of managers and managed objects are well understood in the art of managed networks. Managers and

managed objects have a well-known relationship in managed networks. Shank does not pertain to interactions between managers and managed objects, as these entities are understood in the art. In contrast, Shank only discusses the client-server interactions between application 140 and server 110. In other words, Shank only discusses providing telephony and media services through a server to a client application. Shank does not discuss managing managed network objects. Furthermore, nowhere does Shank teach a query for information concerning a managed object.

In response to the above arguments, the Examiner refers only to Shank's providing services through media, telephony and basic service interfaces, but the Examiner fails to point out anything regarding a *query for information*. The Examiner seems to have misunderstood Appellants' previous argument. Appellants assert that Shank fails to teach wherein the messages *comprise a query for information* concerning one or more of the managed objects. Shank clearly fails to teach anything concerning such a query for information.

Claim 11:

Regarding claim 11, Shank does not disclose wherein the requests comprise a command to set one or more parameters of one of the managed objects. The Examiner cites column 17, lines 53-66 where Shank describes parameters for a specific function (Play) of the Player media service interface, but does not mention a command to set parameters for a managed object. A parameter to a specific service method invocation is very different from a command to set one or more parameters of a managed object and Shank does not mention such a command, just that parameters may be used when invoking specific service method invocations. **The Examiner has failed to ever provide any response to this specific argument when presented previously.**

Claim 13:

In regard to claim 13, Shank fails to anticipate, wherein requests are converted from the interface definition language to a platform-specific format prior to delivery to

the managed objects. The Examiner refers to col. 5, lines 39-50, of Shank. The Examiner's cited portion of Shank discusses examples of media and telephony services but teaches nothing about converting requests from the interface definition language to a platform-specific format prior to delivery to the managed objects. In fact, nowhere does Shank teach or even suggest that requests are converted. Thus, Appellants can see no basis for the Examiner's contention regarding converting requests and assert that the Examiner is merely speculating about Shank's system including converting requests. In response, the Examiner refers to Shank's teachings regarding the communication with different objects by different protocols "based on an industry standard" and further argues, "ASN1 can be implemented in Shank's system." Appellants fail to find any relevance to the Examiner's reference to ASN1. Shank does not mention ASN1 at all and certainly does not teach or suggest the use of ASN1. Furthermore, even if ASN1 were to be implemented in Shank's system, that would not require that Shank's system include the translation of request messages. The Examiner seems to be implying that it would be obvious to modify Shank's system to include the translation of request messages; however, a rejection based on such modification is clearly improper in a § 102(e) anticipation rejection.

The Examiner further argues that Shank teaches "converting requests before delivering them to objects when the user and server [execute] in different process[es]" and cites column 4, lines 35-40 of Shank. However, this portion of Shank discusses the marshalling of a method invocation in order to communicate between a client in one process and a server in a different process. Marshalling of parameters for method invocations does not involve the translation of any request messages. Shank is not discussing translating request messages at all, but rather Shank is discussing parameters used in inter-process communication.

Shank clearly does not anticipate that requests are converted from an interface definition language to a platform-specific format prior to delivery to the managed objects.

Claim 16, 19, 20, 21, 22, 23, 24, 29 and 30:

Shank does not anticipate one or more managers each selecting a format for messages deliverable by a gateway between one or more managed objects and each of the one or more managers. Instead, Shank pertains to providing telephony and media services from a server 110 to an application 140 (Shank, Figure 1, column 1, lines 13-18). Shank is not concerned with, nor does Shank pertain to, managers and managed objects. According to Shank, a server may include various service interfaces, such as telephony services 210, media services 220, and basic services 230 that a client may use. Shank's system provides a CORBA ORB 260 for communicating with these interfaces (col. 3, line 31 - col. 4, line 13). As described in Shank, the service interfaces (such as telephony services 210 and media services 220) allow client application 140 to interact with services such as telephone services provided on telephone network 105 and media services provided by various hardware components (col. 7, lines 15-28).

Contrary to the Examiner's assertions, the service interfaces 210, 220 and 230 of Shank's server 110, do not provide a gateway configurable to deliver the messages for each manager in a format selected by that manager. Shank does not pertain to interactions between managers and managed objects as these entities are understood in the art. Instead, Shank only discusses the client-server interactions between application 140 and server 110.

Please refer to the arguments above regarding claim 1 for a more detailed discussion of Shank's failure to disclose managers selected a desired message format. In short, Shank only discusses providing telephony and media services through a server to a client application. Shank's interfaces 210, 220, 230 provide service interfaces for an application 140. They do not deliver messages between managed objects and one or more managers. Shank clearly does not teach a gateway that is configurable to deliver the messages for each manager in a format selected by that manager. Shank teaches the use of a custom format "based on similar methods specified in the ECTF S 1.00API," but defined using IDL (Shank, column 17, lines 31-34). Thus, Shank clearly teaches the use of predefined message formats and not the use of formats selectable by a manager.

Claim 17:

In regard to claim 17, Shank does not disclose wherein the selected format comprises text, as expressed by the Examiner. The Examiner cites item 228 of Figure 2. However, item 228 refers to a FAX service, and it is well known that a facsimile interface does not include text, but rather involves a graphic interface. Appellants fail to see the relevance of item 228 of Figure 2 to a selected format that comprises *text*.

In response to the above arguments, the Examiner cites Shank's teachings regarding providing text-to-speech services. The text-to-speech services referred to by the Examiner are services accessed by client application 140. For example, the text-to-speech service converts text data supplied by application 140 into an audio file. When discussing text-to-speech, Shank is referring to a high level function performed by the service, not an inter-object message format used for communicating with the service. Even if Shank's application 140 were to represent a manager object, which the appellants assert it does not, Shank still does not teach that application 140 may choose text-to-speech as a format for message delivery when communicating with a service object. In fact, it is clear that when communicating with a text-to-speech service, one must provide the text to be converted into audio. Shank teaches that application 140 must use an interface specified by the text-to-speech service's ORB vendor (Shank, column 3, line 65-column 4, line 6). Thus, Shank's text-to-speech service is not a selectable message format for communicating between a manager and a managed object. Shank clearly does not describe a manager being able to select a format for message delivery that comprises text.

Claim 25:

In regard to claim 25, Shank does not anticipate a gateway comprising a request gateway which is configured to deliver messages generated by the one or more managers to the one or more managed objects, wherein the messages comprises requests and wherein the requests comprise a query for information concerning one of the managed

objects. The portions of Shank cited by the Examiner (column 2, lines 64-67; column 7, lines 43-46; and column 7, line 66 through column 8, line 6) refer to application 140 invoking functions of the telephony and media services. These teachings have nothing to do with a query for information concerning a managed object. The concepts of managers and managed objects are well understood in the art of managed networks. Managers and managed objects have a well-known relationship in managed networks. Shank does not pertain to interactions between managers and managed objects, as these entities are understood in the art. In contrast, Shank only discusses the client-server interactions between application 140 and server 110. In other words, Shank only discusses providing telephony and media services through a server to a client application. Shank does not discuss managing managed network objects. Furthermore, nowhere does Shank teach a query for information concerning a managed object.

In response to the above arguments, the Examiner refers only to Shank's providing services through media, telephony and basic service interfaces, but the Examiner fails to point out anything regarding a *query for information*. The Examiner seems to have misunderstood appellants' previous argument. Appellants assert that Shank fails to teach wherein the messages *comprise a query for information* concerning one or more of the managed objects. Shank clearly fails to teach anything concerning such a query for information.

Claim 26:

Regarding claim 26, Shank does not disclose wherein the requests comprise a command to set one or more parameters of one of the managed objects. The Examiner cites column 17, lines 53-66 where Shank describes parameters for a specific function (Play) of the Player media service interface, but does not mention a command to set parameters for a managed object. A parameter to a specific service method invocation is very different from a command to set one or more parameters of a managed object and Shank does not mention such a command, just that parameters may be used when

invoking specific service method invocations. **The Examiner has failed to ever provide any response to this specific argument when presented previously.**

Claim 28:

In regard to claim 28, Shank fails to anticipate, wherein requests are converted from the interface definition language to a platform-specific format prior to delivery to the managed objects. The Examiner refers to col. 5, lines 39-50, of Shank. The Examiner's cited portion of Shank discusses examples of media and telephony services but teaches nothing about converting requests from the interface definition language to a platform-specific format prior to delivery to the managed objects. In fact, nowhere does Shank teach or even suggest that requests are converted. Thus, appellants can see no basis for the Examiner's contention regarding converting requests and assert that the Examiner is merely speculating about Shank's system including converting requests. In response, the Examiner refers to Shank's teachings regarding the communication with different objects by different protocols "based on an industry standard" and further argues, "ASN1 can be implemented in Shank's system." Appellants fail to find any relevance to the Examiner's reference to ASN1. Shank does not mention ASN1 at all and certainly does not teach or suggest the use of ASN1. Furthermore, even if ASN1 were to be implemented in Shank's system, that would not require that Shank's system include the translation of request messages. The Examiner seems to be implying that it would be obvious to modify Shank's system to include the translation of request messages; however, a rejection based on such modification is clearly improper in a § 102(e) anticipation rejection.

The Examiner further argues that Shank teaches "converting requests before delivering them to objects when the user and server [execute] in different process[es]" and cites column 4, lines 35-40 of Shank. However, this portion of Shank discusses the marshalling of a method invocation in order to communicate between a client in one process and a server in a different process. Marshalling of parameters for method invocations does not involve the translation of any request messages. Shank is not

discussing translating request messages at all, but rather Shank is discussing parameters used in inter-process communication.

Shank clearly does not anticipate wherein requests are converted from an interface definition language to a platform-specific format prior to delivery to the managed objects.

Claim 31, 34, 35, 36, 37, 38, 39, 44 and 45:

Shank does not anticipate one or more managers each selecting a format for messages deliverable by a gateway between one or more managed objects and each of the one or more managers. Instead, Shank pertains to providing telephony and media services from a server 110 to an application 140 (Shank, Figure 1, column 1, lines 13-18). Shank is not concerned with, nor does Shank pertain to, managers and managed objects. According to Shank, a server may include various service interfaces, such as telephony services 210, media services 220, and basic services 230 that a client may use. Shank's system provides a CORBA ORB 260 for communicating with these interfaces (col. 3, line 31 - col. 4, line 13). As described in Shank, the service interfaces (such as telephony services 210 and media services 220) allow client application 140 to interact with services such as telephone services provided on telephone network 105 and media services provided by various hardware components (col. 7, lines 15-28).

Contrary to the Examiner's assertions, the service interfaces 210, 220 and 230 of Shank's server 110, do not provide a gateway configurable to deliver the messages for each manager in a format selected by that manager. Shank does not pertain to interactions between managers and managed objects as these entities are understood in the art. Instead, Shank only discusses the client-server interactions between application 140 and server 110.

Please refer to the arguments above regarding claim 1 for a more detailed discussion of Shank's failure to disclose managers selected a desired message format. In

short, Shank only discusses providing telephony and media services through a server to a client application. Shank's interfaces 210, 220, 230 provide service interfaces for an application 140. They do not deliver messages between managed objects and one or more managers. Shank clearly does not teach a gateway that is configurable to deliver the messages for each manager in a format selected by that manager. Shank teaches the use of a custom format "based on similar methods specified in the ECTF S 1.00API," but defined using IDL (Shank, column 17, lines 31-34). Thus, Shank clearly teaches the use of predefined message formats and not the use of formats selectable by a manager.

Claim 32:

In regard to claim 32, Shank does not disclose wherein the selected format comprises text, as expressed by the Examiner. The Examiner cites item 228 of Figure 2. However, item 228 refers to a FAX service, and it is well known that a facsimile interface does not include text, but rather involves a graphic interface. Appellants fail to see the relevance of item 228 of Figure 2 to a selected format that comprises *text*.

In response to the above arguments, the Examiner cites Shank's teachings regarding providing text-to-speech services. The text-to-speech services referred to by the Examiner are services accessed by client application 140. For example, the text-to-speech service converts text data supplied by application 140 into an audio file. When discussing text-to-speech, Shank is referring to a high level function performed by the service, not an inter-object message format used for communicating with the service. Even if Shank's application 140 were to represent a manager object, which the appellants assert it does not, Shank still does not teach that application 140 may choose text-to-speech as a format for message delivery when communicating with a service object. In fact, it is clear that when communicating with a text-to-speech service, one must provide the text to be converted into audio. Shank teaches that application 140 must use an interface specified by the text-to-speech service's ORB vendor (Shank, column 3, line 65-column 4, line 6). Thus, Shank's text-to-speech service is not a selectable message format for communicating between a manager and a managed object. Shank clearly does

not describe a manager being able to select a format for message delivery that comprises text.

Claim 40:

In regard to claim 40, Shank does not anticipate a gateway comprising a request gateway which is configured to deliver messages generated by the one or more managers to the one or more managed objects, wherein the messages comprises requests and wherein the requests comprise a query for information concerning one of the managed objects. The portions of Shank cited by the Examiner (column 2, lines 64-67; column 7, lines 43-46; and column 7, line 66 through column 8, line 6) refer to application 140 invoking functions of the telephony and media services. These teachings have nothing to do with a query for information concerning a managed object. The concepts of managers and managed objects are well understood in the art of managed networks. Managers and managed objects have a well-known relationship in managed networks. Shank does not pertain to interactions between managers and managed objects, as these entities are understood in the art. In contrast, Shank only discusses the client-server interactions between application 140 and server 110. In other words, Shank only discusses providing telephony and media services through a server to a client application. Shank does not discuss managing managed network objects. Furthermore, nowhere does Shank teach a query for information concerning a managed object.

In response to the above arguments, the Examiner refers only to Shank's providing services through media, telephony and basic service interfaces, but the Examiner fails to point out anything regarding a *query for information*. The Examiner seems to have misunderstood appellants' previous argument. Appellants assert that Shank fails to teach wherein the messages *comprise a query for information* concerning one or more of the managed objects. Shank clearly fails to teach anything concerning such a query for information.

Claim 41:

Regarding claim 41, Shank does not disclose that the requests comprise a command to set one or more parameters of one of the managed objects. The Examiner cites column 17, lines 53-66 where Shank describes parameters for a specific function (Play) of the Player media service interface, but does not mention a command to set parameters for a managed object. A parameter to a specific service method invocation is very different from a command to set one or more parameters of a managed object and Shank does not mention such a command, just that parameters may be used when invoking specific service method invocations. **The Examiner has never provided any response to this specific argument when presented previously.**

Claim 43:

In regard to claim 43, Shank fails to anticipate, wherein requests are converted from the interface definition language to a platform-specific format prior to delivery to the managed objects. The Examiner refers to col. 5, lines 39-50, of Shank. The Examiner's cited portion of Shank discusses examples of media and telephony services but teaches nothing about converting requests from the interface definition language to a platform-specific format prior to delivery to the managed objects. In fact, nowhere does Shank teach or even suggest that requests are converted. Thus, appellants can see no basis for the Examiner's contention regarding converting requests and assert that the Examiner is merely speculating about Shank's system including converting requests. In response, the Examiner refers to Shank's teachings regarding the communication with different objects by different protocols "based on an industry standard" and further argues, "ASN1 can be implemented in Shank's system." Appellants fail to find any relevance to the Examiner's reference to ASN1. Shank does not mention ASN1 at all and certainly does not teach or suggest the use of ASN1. Furthermore, even if ASN1 were to be implemented in Shank's system, that would not require that Shank's system include the translation of request messages. The Examiner seems to be implying that it would be obvious to modify Shank's system to include the translation of request messages; however, a rejection based on such modification is clearly improper in a § 102(e) anticipation rejection.

The Examiner further argues that Shank teaches "converting requests before delivering them to objects when the user and server [execute] in different process[es]" and cites column 4, lines 35-40 of Shank. However, this portion of Shank discusses the marshalling of a method invocation in order to communicate between a client in one process and a server in a different process. Marshalling of parameters for method invocations does not involve the translation of any request messages. Shank is not discussing translating request messages at all, but rather Shank is discussing parameters used in inter-process communication.

Shank clearly does not anticipate wherein requests are converted from an interface definition language to a platform-specific format prior to delivery to the managed objects.

Third Ground of Rejection:

Claims 3, 12, 18, 27, 33 and 42 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Shank. Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Appellants submit that the Examiner has not established a proper *prima facie* case of obviousness in regard to these claims. Obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so in the prior art. *In re Fine*, 837 F.2d 1071 (Fed. Cir. 1988); M.P.E.P. § 2143.01. The question is whether there is something in the prior art as a whole to suggest the desirability, and thus the obviousness, of making the modification or combination. *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 488 (Fed. Cir. 1984). Merely stating that individual aspects of a claimed invention are well known does not render the combination well known without some objective reason to modify the teachings. *Ex parte Levengood*, 28 USPQ2d 1300.

Furthermore, the Examiner's §103(a) rejection amounts to nothing more than pure conclusory speculation by the Examiner. Mere speculation is not sufficient to support a *prima facie* case of obviousness. M.P.E.P. § 2142; *see also, In re Warner*, 379 F.2d 1011, 1017, 154 USPQ 173, 178 (CCPA 1967); *In re Sporck*, 301 F.2d 686, 690, 133 USPQ 360, 364 (CCPA 1962). "The factual inquiry whether to combine references must be thorough and searching." *McGinley v. Franklin Sports, Inc.*, 60 USPQ2d 1001, 1008 (Fed. Cir. 2001). It must be based on objective evidence of record. "This precedent has been reinforced in myriad decisions, and cannot be dispensed with." *In re Sang Su Lee*, 61 USPQ2d 1430 (Fed. Cir. 2002). "The need for specificity pervades this authority." *Id.* "Particular findings must be made as to the reason the skilled artisan, with no knowledge of the claimed invention, would have selected these components for combination in the manner claimed." *In re Kotzab*, 55 USPQ2d 1313, 1317 (Fed. Cir. 2000).

The Examiner has not satisfied the rigorous tests for properly modifying a prior art reference to establish obviousness. Instead, as discussed above, the Examiner's reasoning is not supported by the teachings of the references, lacks specificity, and is based in hindsight.

Claim 3:

Shank fails to teach or suggest wherein the selected format comprises Abstract Syntax Notation One (ASN1). The Examiner has not provided any prior art reference or specific finding that provides a motivation to use ASN.1 in Shank in any way. The Examiner states only that such modifications would be obvious "for fulfilling the system requirements." However, there are no system requirements taught in Shank that would require or even suggest selecting ASN.1 as a message format. The Examiner's stated motivation for modifying Shank ("fulfilling the system requirement") amounts to nothing more than an attempt to build Appellants' invention through hindsight analysis and thus is clearly improper.

Claim 12:

Shank fails to teach or suggest wherein the requests are converted from the interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed objects. The Examiner has not cited any passage of Shank (or any other prior art reference) that teaches or suggests converting requests from IDL to PMI format. Nor has the Examiner presented any motivation to modify Shank to convert requests from interface definition language to a PMI format prior to delivery to the managed objects, as the Examiner contends. Nor is there any teaching in Shank that would require or even suggest converting requests from the interface definition language to a PMI format (or any other format) prior to delivery to the managed objects. As with the rejection of claim 3, discussed above, The Examiner's stated motivation for modifying Shank ("fulfilling the system requirement") amounts to nothing more than an attempt to build the appellants invention through hindsight analysis and is clearly improper. Additionally, Appellants' remarks above regarding the § 102(e) rejection of claim 13 in view of Shank apply here.

Claim 18:

Shank fails to teach or suggest wherein the selected format comprises Abstract Syntax Notation One (ASN1). The Examiner has not provided any prior art reference or specific finding that provides a motivation to use ASN.1 in Shank in any way. The Examiner states only that such modifications would be obvious "for fulfilling the system requirements." However, there are no system requirements taught in Shank that would require or even suggest selecting ASN.1 as a message format. The Examiner's stated motivation for modifying Shank ("fulfilling the system requirement") amounts to nothing more than an attempt to build the appellants invention through hindsight analysis and thus is clearly improper.

Claim 27:

Shank fails to teach or suggest wherein the requests are converted from the interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed objects. The Examiner has not cited any passage of Shank (or any other prior art reference) that teaches or suggests converting requests from IDL to PMI format. Nor has the Examiner presented any motivation to modify Shank to convert requests from interface definition language to a PMI format prior to delivery to the managed objects, as the Examiner contends. Nor is there any teaching in Shank that would require or even suggest converting requests from the interface definition language to a PMI format (or any other format) prior to delivery to the managed objects. As with the rejection of claim 3, discussed above, The Examiner's stated motivation for modifying Shank ("fulfilling the system requirement") amounts to nothing more than an attempt to build the appellants invention through hindsight analysis and is clearly improper. Additionally, Appellants' remarks above regarding the § 102(e) rejection of claim 28 apply here.

Claim 33:

Shank fails to teach or suggest wherein the selected format comprises Abstract Syntax Notation One (ASN1). The Examiner has not provided any prior art reference or specific finding that provides a motivation to use ASN.1 in Shank in any way. The Examiner states only that such modifications would be obvious "for fulfilling the system requirements." However, there are no system requirements taught in Shank that would require or even suggest selecting ASN.1 as a message format. The Examiner's stated motivation for modifying Shank ("fulfilling the system requirement") amounts to nothing more than an attempt to build the appellants invention through hindsight analysis and thus is clearly improper.

Claim 42:

Further regarding claim 42, Shank fails to disclose wherein the requests are converted from the interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed objects. The Examiner provided any prior

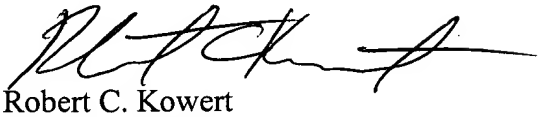
art reference or specific finding that provides a motivation to modify Shank to convert requests from the interface definition language to a PMI format prior to delivery to the managed objects, as the Examiner contends. Nor is there any teaching in Shank that would require or even suggest converting requests from the interface definition language to a PMI format prior to delivery to the managed objects. As with the rejection of claim 3, discussed above, The Examiner's stated motivation for modifying Shank ("fulfilling the system requirement") amounts to nothing more than an attempt to build the appellants invention through hindsight analysis and is clearly improper. Additionally, Appellants' remarks above regarding the § 102(e) rejection of claim 43 in view of Shank apply here.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-45 was erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of $500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-61100/RCK. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,



Robert C. Kowert
Reg. No. 39,255
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: February 18, 2005

IX. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A network management system comprising:

a gateway which is coupled to one or more managed objects and which is configured to deliver messages between the managed objects and one or more managers; and

a platform-independent interface to the gateway, wherein the gateway is configurable to communicate with the managers through the platform-independent interface to deliver the messages;

wherein the gateway is configurable to deliver the messages for each manager in a format selected by that manager.

2. The network management system of claim 1, wherein the selected format comprises text.

3. The network management system of claim 1, wherein the selected format comprises Abstract Syntax Notation One (ASN1).

4. The network management method of claim 1, wherein the messages are communicated with the managers via Internet Inter-Object Protocol (IIOP).

5. The network management system of claim 1, wherein the platform-independent interface to the gateway is expressed in an interface definition language, and wherein the interface definition language comprises a language for defining interfaces to managed objects across a plurality of platforms and across a plurality of programming languages.

6. The network management system of claim 5, wherein the interface definition language comprises OMG IDL.

7. The network management system of claim 1, wherein the managed objects comprise one or more objects corresponding to a telephone network.

8. The network management system of claim 1, wherein the managed objects comprise an object corresponding to a telecommunications device.

9. The network management system of claim 1, wherein the gateway comprises a request gateway which is configured to deliver messages generated by the one or more managers to the one or more managed objects, and wherein the messages comprise requests for the one or more managed objects.

10. The network management system of claim 9, wherein the requests comprise a query for information concerning one of the managed objects.

11. The network management system of claim 9, wherein the requests comprise a command to set one or more parameters of one of the managed objects.

12. The network management system of claim 9, wherein the requests are converted from the interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed objects.

13. The network management system of claim 9, wherein the requests are converted from the interface definition language to a platform-specific format prior to delivery to the managed objects.

14. The network management system of claim 1, wherein the gateway comprises an event gateway, and wherein the messages comprise events associated with

the managed objects.

15. The network management system of claim 14, the events comprise an alert generated by one of the managed objects.

16. A network management method, comprising:

one or more managers each selecting a format for messages deliverable by a gateway between one or more managed objects and each of the one or more managers, wherein the gateway is configurable to communicate with the managers through a platform-independent interface to deliver the messages; and

delivering the messages between the one or more managed objects and the one or more managers, according to the format selected by each manager.

17. The network management method of claim 16, wherein the selected format comprises text.

18. The network management method of claim 16, wherein the selected format comprises Abstract Syntax Notation One (ASN1).

19. The network management method of claim 16, wherein the messages are communicated with the managers via Internet Inter-Object Protocol (IIOP).

20. The network management method of claim 16, wherein the platform-independent interface to the gateway is expressed in an interface definition language, and wherein the interface definition language comprises a language for defining interfaces to managed objects across a plurality of platforms and across a plurality of programming languages.

21. The network management method of claim 20, wherein the interface

definition language comprises OMG IDL.

22. The network management method of claim 16, wherein the managed objects comprise one or more objects corresponding to a telephone network.

23. The network management method of claim 16, wherein the managed objects comprise an object corresponding to a telecommunications device.

24. The network management method of claim 16, wherein the gateway comprises a request gateway which is configured to deliver messages generated by the one or more managers to the one or more managed objects, and wherein the messages comprise requests for the one or more managed objects.

25. The network management method of claim 24, wherein the requests comprise a query for information concerning one of the managed objects.

26. The network management method of claim 24, wherein the requests comprise a command to set one or more parameters of one of the managed objects.

27. The network management method of claim 24, wherein the requests are converted from the interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed objects.

28. The network management method of claim 24, wherein the requests are converted from the interface definition language to a platform-specific format prior to delivery to the managed objects.

29. The network management method of claim 16, wherein the gateway comprises an event gateway, and wherein the messages comprise events associated with the managed objects.

30. The network management method of claim 29, the events comprise an alert generated by one of the managed objects.

31. A carrier medium comprising program instructions for network management, wherein the program instructions are computer-executable to perform:

one or more managers each selecting a format for messages deliverable by a gateway between one or more managed objects and the one or more managers, wherein the gateway is configurable to communicate with the managers through a platform-independent interface to deliver the messages; and

delivering the messages between the one or more managed objects and the one or more managers.

32. The carrier medium of claim 31, wherein the selected format comprises text.

33. The carrier medium of claim 31, wherein the selected format comprises Abstract Syntax Notation One (ASN1).

34. The carrier medium of claim 31, wherein the messages are communicated with the managers via Internet Inter-Object Protocol (IIOP).

35. The carrier medium of claim 31, wherein the platform-independent interface to the gateway is expressed in an interface definition language, and wherein the interface definition language comprises a language for defining interfaces to managed objects across a plurality of platforms and across a plurality of programming languages.

36. The carrier medium of claim 35, wherein the interface definition language comprises OMG IDL.

37. The carrier medium of claim 31, wherein the managed objects comprise one or more objects corresponding to a telephone network.

38. The carrier medium of claim 31, wherein the managed objects comprise an object corresponding to a telecommunications device.

39. The carrier medium of claim 31, wherein the gateway comprises a request gateway which is configured to deliver messages generated by the one or more managers to the one or more managed objects, and wherein the messages comprise requests for the one or more managed objects.

40. The carrier medium of claim 39, wherein the requests comprise a query for information concerning one of the managed objects.

41. The carrier medium of claim 39, wherein the requests comprise a command to set one or more parameters of one of the managed objects.

42. The carrier medium of claim 39, wherein the requests are converted from the interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed objects.

43. The carrier medium of claim 39, wherein the requests are converted from the interface definition language to a platform-specific format prior to delivery to the managed objects.

44. The carrier medium of claim 31, wherein the gateway comprises an event gateway, and wherein the messages comprise events associated with the managed objects.

45. The carrier medium of claim 44, the events comprise an alert generated by

one of the managed objects.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.